GETTEXT

Susceptible to environment variable driven buffer overflows

Sean Barnum, Cigital, Inc. [vita[1]]

Copyright © 2007 Cigital, Inc.

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5438 bytes

<table>
<tr><td><b>Attack Category</b></td><td>• Environment Manipulation</td></tr>
<tr><td><b>Vulnerability Category</b></td><td>• Buffer Overflow
• Unconditional</td></tr>
<tr><td><b>Software Context</b></td><td>• National Language Support
• String Conversion MACROS</td></tr>
<tr><td><b>Location</b></td><td>• libintl.h</td></tr>
<tr><td><b>Description</b></td><td>The gettext function attempts to translate a text string into the user's native language by looking up the translation in a message catalog. The msgid argument identifies the message to be translated. By convention, it is the English version of the message, with non-ASCII characters replaced by ASCII approximations. This choice allows the translators to work with message catalogs, called PO files, that contain both the English and the translated versions of each message and can be installed using the msgfmt utility.
A message domain is a set of translatable msgid messages. Usually, every software package has it own message domain. The domain name is used to determine the message catalog where the translation is looked up; it must be a non-empty string. For the gettext function, it is specified through a preceding textdomain call.
gettext() accesses unvalidated information from the environment. Similar to getcat(), this function is used to look up messages in a message catalog, presumably for natural language translation.
gettext() is susceptible to environment variable driven buffer overflows.</td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>dcgettext</td><td></td></tr></table></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

dcngettext	returns arbitrary length string
dgettext	
dngettext	returns arbitrary length string
gettext	
ngettext	returns arbitrary length string

Method of Attack

An attacker can manipulate the environment (LC_ALL or LC_MESSAGES environment variables) to provide a malicious set of messages that might overrun buffers. Static buffers need to be used with care.

Exception Criteria

Solutions

Solution Applicability	Solution Description	Solution Efficacy
Whenever using gettext()	All output from gettext() should be validated. If possible, validate LC_ALL and LC_MESSAGES prior to calling it. gettext() returns a const char* to the message; length should be checked before copying it to any buffer.	Effective.

Signature Details

char * gettext (const char * msgid);

Examples of Incorrect Code

```
char * translation =
gettext("Please try again");
char * error_msg [5];
strcpy(error_msg, translation);
```

Examples of Corrected Code

```
int buf_len = 10;
char * translation =
gettext("Please try again");
char * error_msg [buf_len];
strncpy(error_msg, translation,
buf_len);
```

Source References

- Rough Auditing Tool for Security (RATS)[2]
- gettext() man page
- GETTEXT [3](2001).

<table>
<tr><td></td><td><ul><li>Arce, Ivan. FOLLOUP: UNIX locale vulnerability[4] (2000).</li></ul></td></tr>
<tr><td><b>Recommended Resource</b></td><td></td></tr>
<tr><td><b>Discriminant Set</b></td><td><table><tr><td><b>Operating System</b></td><td><ul><li>Windows</li></ul></td></tr><tr><td><b>Languages</b></td><td><ul><li>C</li><li>C++</li></ul></td></tr></table></td></tr>
</table>

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com[1].

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com